

Contents

1 Routine/Function Prologues	2
1.0.1 gethuff.F90 (Source File: gethuff.F90)	2
1.0.2 hufffile (Source File: gethuff.F90)	5

1 Routine/Function Prologues

1.0.1 gethuff.F90 (Source File: gethuff.F90)

Opens and reads global precipitation forcing

CTIME = Current time

FTIMENRL = Nearest future data for NRL data

FTIMEHUFF = Nearest future data for HUFFMAN data

FTIMEPERS = Nearest future data for PERSIANN data

REVISION HISTORY:

17 Jul 2001: Jon Gottschalck; Initial code

10 Oct 2001: Jon Gottschalck; Modified to adjust convective precip
using a ratio of the model convective / total ratio

30 Jul 2002: Jon Gottschalck; Added PERSIANN and HUFFMAN global observed precip data sources

INTERFACE:

```
subroutine gethuff
```

USES:

```
use lisdrv_module, only : lis, gindex
use time_manager
use time_module, only : tick,time2date
use huffdomain_module, only : huffdrv
implicit none
```

CONTENTS:

```
!-----
! Set parameter to measure 1.5 hour time offset when using HUFFMAN
!-----
gap = 0.0001712328767098370
!-----
! Determine required observed precip data times
! (current, accumulation end time)
! Model current time
!-----
yr1 = lis%t%yr !current time
mo1 = lis%t%mo
da1 = lis%t%da
hr1 = lis%t%hr
mn1 = lis%t%mn
ss1 = 0
ts1 = 0
call tick( ctime, doy1, gmt1, yr1, mo1, da1, hr1, mn1, ss1, ts1 )
```

```
!-----
```

```

! HUFFMAN product end time
!-----
yr3 = lis%t%yr !end accumulation time data
mo3 = lis%t%mo
da3 = lis%t%da
hr3 = 3*(lis%t%hr/3)
mn3 = 0
ss3 = 0
ts3 = 3*60*60
call tick( ftime_huff, doy3, gmt3, yr3, mo3, da3, hr3, mn3, ss3, ts3 )
breaktime = ftime_huff - ctime
datatime = ftime_huff
fnametime = ftime_huff
if (lis%f%gpcsrc == 3) then
  if (breaktime .ge. gap) then
    call time2date( datatime, kdoy3, kgmt3, kyr3, &
      kmo3, kda3, khr3, kmn3 )
    call time2date( fnametime, mdoy3, mgmt3, myr3, &
      mmo3, mda3, mhr3, mmn3 )
    flag1 = 1
    if (khr3 == 24) khr3 = 0
    if (mhr3 == 24) mhr3 = 0
    if (kgmt3 .eq. 0.0 .and. flag2 .eq. 2) then
      kts3 = -25.5*60*60
      call tick( datatime, kdoy3, kgmt3, kyr3, kmo3, &
        kda3, khr3, kmn3, kss3, kts3 )
      mts3 = -27*60*60
      call tick( fnametime, mdoy3, mgmt3, myr3, mmo3, &
        mda3, mhr3, mmn3, mss3, mts3 )
    else
      kts3 = -1.5*60*60
      call tick( datatime, kdoy3, kgmt3, kyr3, kmo3, &
        kda3, khr3, kmn3, kss3, kts3 )
      mts3 = -3*60*60
      call tick( fnametime, mdoy3, mgmt3, myr3, mmo3, &
        mda3, mhr3, mmn3, mss3, mts3 )
    endif
    flag2 = 1
  else
    if (get_nstep().eq. 1) then
      call time2date( datatime, kdoy3, kgmt3, kyr3, kmo3, &
        kda3, khr3, kmn3 )
      call time2date( fnametime, mdoy3, mgmt3, myr3, mmo3, &
        mda3, mhr3, mmn3 )
      if (kgmt3 .eq. 0) then
        mts3 = -24*60*60
        call tick( fnametime, mdoy3, mgmt3, myr3, mmo3, &
          mda3, mhr3, mmn3, mss3, mts3 )
      endif
    endif
  endif
endif

```

```

        kts3 = -22.5*60*60
        call tick( datatime, kdoy3, kgmt3, kyr3, kmo3, &
                   kda3, khr3, kmn3, kss3, kts3 )
    else
        mts3 = 0
        call tick( fnametime, mdoy3, mgmt3, myr3, mmo3, &
                   mda3, mhr3, mmn3, mss3, mts3 )
        kts3 = 1.5*60*60
        call tick( datatime, kdoy3, kgmt3, kyr3, kmo3, &
                   kda3, khr3, kmn3, kss3, kts3 )
    endif
    else
        flag1 = 2
        if (flag2 .eq. 1) then
            mts3 = 3*60*60
            call tick( fnametime, mdoy3, mgmt3, myr3, mmo3, &
                       mda3, mhr3, mmn3, mss3, mts3 )
            kts3 = 3*60*60
            call tick( datatime, kdoy3, kgmt3, kyr3, kmo3, &
                       kda3, khr3, kmn3, kss3, kts3 )
        endif
        flag2 = 2
    endif
endif
!-----
! Ensure that data is found during first time step
!-----
if ( lis%f%gpcpsrc.eq.3 .and. get_nstep().eq. 1 ) endtime_huff = 1
!-----
! Check for and get HUFFMAN observed Precipitation data
!-----
if (lis%f%gpcpsrc.eq.3) then
    if ( ctime > huffdrv%hufftime ) then
        endtime_huff = 1
        if ( endtime_huff == 1 ) then !get new time2 data
            print*, 'Getting new HUFFMAN satellite precip data', endtime_huff
            ferror_huff = 0
            call hufffile( name, huffdrv%huffdir, myr3, mmo3, mda3, mhr3 )
            call glbprecip_huff( name, ferror_huff )
            huffdrv%hufftime = datatime
        endif
    endif
endif
return

```

1.0.2 hufffile (Source File: gethuff.F90)

This subroutine puts together HUFFMAN file name for 3 hour file intervals

INTERFACE:

```
subroutine hufffile( name, huffdir, yr, mo, da, hr)
```

CONTENTS:

```
92 format (80a1)
93 format (a80)
94 format (i4, i2, i2, i2)
95 format (10a1)
96 format (a40)
97 format (a4)
98 format (a1, i4, i2, a1)
99 format (8a1)
89 format (a7)
```

```
!-----
! Make variables for the time used to create the file
! We don't want these variables being passed out
!-----
```

```
uyr = yr
umo = mo
uda = da
uhr = 3*(hr/3) !hour needs to be a multiple of 3 hours
umn = 0
uss = 0
ts1 = -24*60*60 !one day interval to roll back date.
```

```
open(unit=90, file='temp', form='formatted', access='direct', recl=80)
write(90, 96, rec=1) huffdir
read(90, 92, rec=1) (fbase(i), i=1,80)
```

```
write(90, 98, rec=1) '/', uyr, umo, '/'
read(90, 99, rec=1) fdir
do i = 1, 8
  if ( fdir(i) == ' ') fdir(i) = '0'
end do
```

```
write(90, 94, rec=1) uyr, umo, uda, uhr
read(90, 95, rec=1) ftime
do i = 1, 10
  if ( ftime(i) == ' ') ftime(i) = '0'
end do
```

```
write(90, 97, rec=1) '.bin'
read (90, 92, rec=1) (fsubs(i), i=1,4)

write(90, 89, rec=1) '3B42RT.'
read (90, 92, rec=1) (fprefix(i), i=1,7)
c = 0
do i = 1, 80
  if ( (fbase(i) == ' ') .and. (c == 0) ) c = i-1
end do

write(90, 92, rec=1) (fbase(i),i=1,c),(fdir(i),i=1,8),(fprefix(i), i=1,7),  &
(fftime(i), i=1,10), (fsubs(i), i=1,4)

read(90, 93, rec=1) name

close(90)
return
```